REMARKS

Claims 14-34 are pending.

Claims 14-33 are rejected under 35 USC 112, second paragraph, for being indefinite as indicated. The claims are amended taking into consideration the Examiner comments. Withdrawal of the indefiniteness rejection is respectfully requested.

Claims 14, 15-20 and 33 are rejected under 35 USC 102(b) as being anticipated by Krauskopf (US Patent No. 5,165,027).

Claims 21-32 are rejected under 35 USC 102(e) as being anticipated by Alverson (US Patent No. 6,480,818).

Claim 34 is rejected under 35 USC 102(e) as being anticipated by Alpert (US Patent No. 5,740,413).

The claims are amended, and, thus, the pending claims remain pending for reconsideration, which is respectfully requested. No new matter has been added.

PRIOR ART REJECTIONS

The independent claims are 14, 15, 21, 27 and 33.

The Examiner maintains from the previous office action the anticipation rejection of independent claims 14, 15 and 33 over Krauskopf.

The Office Action page 20 is the Response to Arguments, in which the Examiner maintains a 'breakpoint instruction' (or software break) is same as the claimed "conditional instruction," because when a breakpoint instruction is executed processing is shifted to an interrupt handler, which is allegedly same as branching to a designated data processing.

However, a 'breakpoint instruction' is an instruction that generates an interrupt, while a "conditional instruction" branches to another destination instruction. One skilled in the art recognizes the difference between a "conditional instruction" and a 'breakpoint instruction' as discussed in the present application pages 7-8. The concept of an interrupt to an interrupt handler differs from a branch to another destination instruction. In other words, as discussed in the present application page 15, lines 12-20 and page 20, lines 12 to page 23, line 17, a "conditional instruction" is an instruction for which it is determined whether a condition is satisfied, and only when the condition is satisfied a designated processing is performed. In contrast, a 'breakpoint instruction' is an instruction which always causes an interrupt to an

interrupt handler regardless of whether a condition is satisfied. In other words, the language of the claims provides "a condition determination section for determining whether a branch condition of ~~a branch of~~ said conditional instruction is satisfied." In other words, a 'breakpoint instruction' does not have a branch (i.e., a break point instruction does not have the claim language "when a designated branch condition of ~~a branch~~ the conditional instruction is satisfied").

Further, the language of the claims provides both break detection and a condition determination section for determining a branch condition of a conditional instruction, clarifying that 'instruction breakpoints or breakpoint instructions/software breakpoints' differs from a "conditional instruction" and cannot be the same. In other words, the language of the claims differentiates a 'breakpoint' from "a conditional instruction" that has a "branch condition," such that the Office Action interpretation that a 'breakpoint' is same type of instruction as a conditional instruction having a branch condition would be contrary to knowledge of one skilled in the art. In other words, the claims clearly differentiate a 'breakpoint' from "a conditional instruction" by reciting "a conditional instruction that executes a designated data processing when a designated branch condition of ~~a branch~~the conditional instruction is satisfied, wherein ~~and~~ a determination of the branch condition of the ~~branch~~conditional instruction and the executed data processing when the branch condition of the conditional instruction is satisfied are indivisible."

The independent claims 14, 15, 21, 27, 33 and 34 are amended. See, for example, embodiments 10-22. According to an aspect of an embodiment, whether to cause an interrupt according to any breakpoint (e.g., instruction breakpoint, breakpoint instruction/software breakpoint, etc.), depends upon "when a designated branch condition of the conditional instruction~~branch~~ is satisfied."

Krauskopf is silent on any concept of conditional breakpoints. Further, Krauskopf does not discuss any "conditional instruction that executes a designated data processing when a designated branch condition of ~~a branch~~the conditional instruction is satisfied," which differs from both types of instruction breakpoints and breakpoint instruction/software breakpoints. The claimed "a condition determination section for determining whether a branch condition of ~~a branch of~~ said conditional instruction is satisfied" determines whether a branch condition of the conditional instruction which is a branch condition of an instruction written/set in a computer program is satisfied. On the other hand, Krauskopf merely discusses, as described in column 4, lines 15-39 and column 5, lines 33-46, that component 36, 32 shown in Fig. 2 is for determining

whether a condition of a breakpoint matches, especially, for determining whether a bus cycle is a reference to computer program or data, and in the case of data references, whether the bus cycle is a read or a write cycle. As explained above, a prima facie case of anticipation based upon Krauskopf cannot be established, because Krauskopf's 'component 36, 32' are completely different from the claimed embodiment by failing to disclose expressly or inherently the claimed "a condition determination section for determining whether a branch condition of ~~a branch of~~ said conditional instruction is satisfied."

Further, none of the other cited references discloses any component corresponding to the claimed "a condition determination section for determining whether a branch condition of ~~a branch of~~ said conditional instruction is satisfied," as follows:

Independent claims 21 and 27 are rejected under 35 USC 102(e) as being anticipated by Alverson (US Patent No. 6,480,818). Alverson column 21, lines 51+ discuss a breakpoint handler retrieving information from the nub about the breakpoint, such as whether the breakpoint is conditional and valid. However, in Alverson a conditional breakpoint merely refers to whether the nub has designated the breakpoint as valid or invalid, which differs from while executing a "conditional instruction that executes a designated data processing when a designated branch condition of ~~a branch~~the conditional instruction is satisfied," detecting an interrupt and "a control section for, in an interrupt handler activated in accordance with said break-interrupt notification supplied from said instruction break detection section, determining whether a branch condition of ~~a branch of~~ said conditional instruction is satisfied, and controlling break-interrupt processing in accordance with the determining of the branch condition of the conditional instruction."

Independent claim 34 is rejected under 35 USC 102(e) as being anticipated by Alpert (US Patent No. 5,740,413). Alpert column 6, lines 38-55, which is relied upon by the Office Action, discusses a branch breakpoint unit 190 including circuitry detecting whether an instruction currently being executed is causing a branch. Further, Alpert column 6, lines 48-52 discuss:

> A branch breakpoint event is not generated for conditional-branch instructions that do not result in a branch being taken. While enable bit 152 indicates the enable state, branch breakpoint unit 190 transmits a signal each time it detects a branch is or will not be taken … upon [which] execution unit 142 recognizes a branch breakpoint event.

In Alpert the enable bit 152 is in the status register 150. Alpert does not discuss details of Alpert's branch breakpoint unit 190. Alpert cannot anticipate the claimed embodiment by

failing to disclose expressly or inherently "determining a branch of an instruction; and controlling a break-interrupt based upon the detecting the breakpoint and the determining of the branch of the instruction" and "controlling a break-interrupt based upon the detecting the breakpoint and the determining of the branch of the instruction, according to a logical operation of a detection signal from said breakpoint detection and a branch condition determination signal from said branch condition determination of the instruction; and sending a break-interrupt notification in accordance with the logical operation."

Independent claims 14, 15, 33, and 34 patentably distinguish over Krauskopf, Alverson and Alpert, and withdrawal of the rejection of pending claims and allowance of pending claims is respectfully requested.

CONCLUSION

If there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 19-3935.

Respectfully submitted,
STAAS & HALSEY LLP

Date: October 16, 2007

By: / Mehdi D. Sheikerz /
Mehdi D. Sheikerz
Registration No. 41,307

1201 New York Ave, N.W., 7th Floor
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501